

OpenMP Tasks

1. 100.000.000 Natural Numbers

Write a program "sum.c" which calculates the sum of the first 100,000,000 natural numbers.

- Distribute the work over 10 pthreads.
- What does the same program look like with OpenMP?
- Is a solution in which sections or tasks are used possible and sensible? Justify in writing (max 1/2 page) and/or add code/functions to your program to justify.

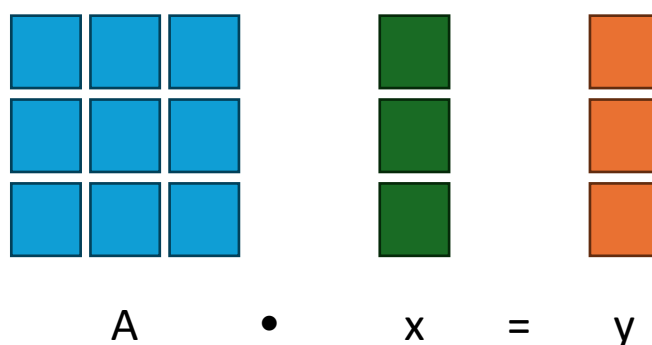
2. Sum of all Values in an Array

The sum of an array consisting of random values is to be calculated. The size of the array should be able to be flexibly defined via "define" at the start of the program.

- Write a serial program "array_sum_serial.c" which performs the addition and outputs a result.
- Parallelize your program from task a with OpenMP. Name the new program "array_sum_parallel.c".
- Is a solution in which sections or tasks are used possible and sensible? Justify in writing (max 1/2 page) and/or add code/functions to your program to justify.

3. Matrix-Vector Multiplication

Implement a matrix-vector multiplication. To simplify things, you can assume a square matrix ($n \times n$), whereby the vector then consists of n columns. The dimension n should be changeable via "#define".



- Write a serial program "matrix_vector_serial.c" which performs the multiplication and outputs a result.
- Parallelize your program from task a with OpenMP. Name the new program "matrix_vector_parallel.c".
- Run your programs with matrices/vectors of different sizes and measure the runtimes. Provide a corresponding table or graph representing the runtimes and explain the program behavior (max. 1/2 page).